Quality and Testing Measures and Metrics

- Develop a measurement and metrics plan tailored to your organization
- Select the right measures for your project and organization
- Explore testing's role in software measurement
- Create a custom quality and test metrics dashboard
- Learn how to avoid dysfunctional metrics

To be most effective, leaders—such as development and testing managers, ScrumMasters, product owners, and IT managers—need metrics to help direct their efforts and make informed recommendations about the software's release readiness and associated risks. Because one important evaluation activity is to "measure" the quality of the software, the progress and results of both development and testing must be measured. Collecting, analyzing, and using metrics are complicated because developers and testers often are concerned that the metrics will be used against them.

In this course, you will create your own metrics plan, establish the guidelines for developing a quality and test measurement program, identify rules of thumb for metrics information, and learn how to avoid "metrics dysfunction." This course addresses common metrics—measures of product quality, defect removal efficiency, defect density, defect arrival rate, testing status, and more.

The Leader's Role in Measurement
Providing timely and accurate quality and release readiness information to project stakeholders is one of the most important values of testing. As a byproduct of testing efforts, the team(s) and team leaders need to continually measure and report the status and quality of the products and features under development. At the same time, they should measure test effectiveness as a guide for improving testing practices. The team and its leaders must determine when to stop testing and fixing bugs—and release the product. Because a key component of testing is measuring the quality of the software product, team leaders and other key stakeholders need timely information related to the entire software development activity.

Who Should Attend?
This course provides foundational information on planning, selecting, and implementing metrics and measures for your team or organization. The course walks you through the creation of a measurement and metrics plan tailored to your organization. Anyone responsible for defining and reporting process and product measures for testing efforts—software and test managers; project managers; ScrumMasters; development or test leaders, those in testing roles, and QA personnel—can benefit from this course.

Coveros recommends this course to help prepare you forAdvanced Tester Certification—Test Managers course[1] and certification exam.

Course Outline

Course Overview
Course summary
Course goals and outline
Instructor introduction

Course Context

A Quality and Test Dashboard
Quality of product
Test status
Test effectiveness
Test efficiency
Resources

Software as a series of imperfect translations
What is software measurement

Common and Uncommon Measures
Primitive and computed measures and metrics
Common metrics
Uncommon metrics

Basic Definitions of Software Measurement
Measure, metric, meter, meta-measure
What makes a good measure or metric

Why Measure?
Understand progress
Determine readiness and results
Take corrective action
Drive improvement
Get high quality products to market more quickly and at the right cost

Measurement and Metrics— Challenges and Requirements
Measurement challenges
Measures and metrics key requirements
Key principles

Metrics Rules of Thumb
The human element
The basics
KISS
And a myth or two

Foundational Material for Your Measurement and Metrics Plan
Templates
Questions
Examples
Virtual Consulting included with this course

Define Your Measurement Plan
Measurement & metrics plan—example
Define with your end in mind
Engage others

Assess Your Current Measures and Metrics
Measures and metrics inventory—example
Gap analysis
Measurement and Metrics— Composition
Engage key stakeholders
Issues

The Quality of Product Gauge
Functional defect information
Non-functional defect information
Defect clustering
Code complexity

The Test Status Gauge
Test execution information
Defect information
Coverage information
Risk information

The Test Effectiveness Gauge
Coverage information
Defect information
Customer feedback
Defect cost
Traceability
Defect detection percentage

The Test Efficiency Gauge
Time required to design, equip, or execute test cases
Defect removal efficiency
Time to run tests
Time to develop and maintain automated tests
Time to equip and deploy environments

The Resources Gauge
Actual vs estimated for test planning and budgeting
Staffing—internal or external
Test environment utilization
Other hardware/software resources

The Issues Gauge
Commentary and narrative
Roadblocks, constraints, risks, skill set needs, training, availability of new hardware or software, etc.

Using the Dashboard
Avoiding dysfunction
Truths and myths
Collecting and Managing Metrics Information
Obtaining metrics information
Measurement without data
Tools to support measurement and metrics

Selling, Piloting, and Implementing Metrics

Which stakeholders are important for your plan

Measures and Metrics in the Context of Your Development Lifecycle
Agile and metrics
Waterfall and metrics
Hybrid lifecycles and metrics

Goal, Question, Metric + Strategies Paradigm
Link goals to metrics
GQM + Strategies—example
Selling metrics—qualitative questions
Selling metrics—quantitative questions

Lessons Learned and Wrap-Up
Advice on things to do and not to do
Keys to success
Course summary